

### Remarks

This Request for Continued Examination and Reply is in response to the Office Action mailed December 12, 2008.

#### **I. Summary of Examiner's Rejections**

In the Office Action mailed December 12, 2008, Claims 50-63 and 71-75 were rejected under 35 U.S.C. 103(a) as being unpatentable over "Event-Oriented Dynamic Adaptation of Workflows: Model, Architecture, and Implementation" (hereinafter Muller) in view of "Web Service Orchestration" (HP, 1-2003, hereinafter Peltz).

#### **II. Summary of Applicants' Amendments**

The present Reply amends Claims 50, 57 and 75 and adds Claims 77-81, leaving for the Examiner's present consideration Claims 50-81.

#### **III. Claim Rejections under 35 U.S.C. §103(a)**

In the Office Action, Claims 50-63 and 71-75 were rejected under 35 U.S.C. 103(a) as being unpatentable over Muller in view of Peltz.

#### **Claim 50**

Claim 50 has been amended to more clearly define the embodiment therein. As amended, Claim 50 defines:

50. *A system for utilizing a workflow language, comprising:*  
*a computer including a processing device operating thereon;*  
*a program source file stored on a computer readable medium, wherein the program source file includes a source code and classes therein;*  
*a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes, and wherein said workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein, and wherein the workflow definition further includes a construct to terminate the parallel processing of the workflow when certain conditions are met; and*  
*object code executed by the processor, the object code configured to*  
*use a workflow program according to said workflow definition, including processing, using the computer including the processing device operating thereon, the plurality of workflow constructs to activate a workflow, including creating separate workflow processes corresponding to the separate workflow branches,*  
*pass information including files, documents, and/or tasks between system resources according to a set of procedural rules to generate activities at the computer as*

- 8 -

Application No. 10/784,375  
Attorney Docket No. ORACL-01389US1  
M:\ahipp\wp\ORACL\1389US1\1389US1\_ROA\_03052009\_2.doc

*defined by the workflow definition specified in the form of annotations to the source code and classes,*

*activate each of the separate workflow processes to subsequently generate activities at the computer as defined by each workflow branch, and*

*determine when the certain conditions specified in the source file have occurred and then terminating the parallel processing of the workflow.*

As currently amended, Claim 50 defines using a workflow language “in the form of annotations to the source code and the classes” to enable developers that are used to working with a particular source code to add workflow definitions in the form of annotations to the source code to “pass information including files, documents, and/or tasks between system resources according to a set of procedural rules to generate activities at the computer as defined by the workflow definition specified in the form of annotations to source code.”

In a traditional system, such annotation would normally be ignored when the source code is processed, because annotations normally are not part of the program itself, and they normally have no direct effect on the operation of the source code they annotate. However, as defined by amended Claim 50, such annotations are read by the system and are used to create a workflow program that passes information between system resources according to a set of procedural rules such that the system acts upon the information, rather than being ignored.

Muller discloses an agent-based workflow management system that can handle various failure situations. As disclosed therein, the AGENTWORK system supports the definition, the execution and, as its main contribution, the event-oriented and semi-automated dynamic adaption of workflows (page 1). Figure 1-2 of Muller (at page 5) illustrates a simplified architecture of a workflow management system, which includes a workflow definition.

In the Office Action, it was acknowledged that Muller does not explicitly address that the program source file includes a source code and classes therein, and workflow definition created using a workflow language that is specified in the form of annotations to the source code. However, it was asserted that Peltz discloses that a workflow definition is invoked using WSFL where WSFL includes Java, i.e., program source file that includes source code and classes therein, and JavaDoc annotation tags in the workflow language that is specified in the form of annotations to the source code and classes.

Applicant respectfully disagrees that the features of Claim 50 are obvious in view of Peltz. In particular, it was submitted that Peltz discloses that a workflow definition can be invoked using Web Services Flow Language (WSFL) (page 5 of Peltz), and that certain process logic can be written using Java program source code which includes program source code and classes. Based on the above description, it appears that in Peltz, a workflow definition is invoked using WSFL or Java program source

code. As amended, Claim 50 defines “a workflow definition created using a workflow language that is specified in the form of annotations to the source code and the classes where the workflow language in the form of annotations to the source code and the classes passes information including files, documents, and/or tasks between system resources according to a set of procedural rules to generate activities at the computer as defined by the workflow definition.” Thus, while Peltz discloses the use of WSFL or Java program source code to accomplish a process flow, the embodiments of Claim 50 define a workflow language in the form of annotations to the source code to generate activities at the computer. As such, Applicant respectfully submits that Peltz does not disclose that annotations to source code and classes can be used to generate an activity at the computer, as defined by amended Claim 50.

Furthermore, it was asserted in the Office Action that JavaDoc annotation tags in the workflow language are specified in the form of annotations to the source code and classes (Office Action, page 6). For example, to start a conversation, a developer would include the tag, “@jws:conversation: phase=start” (page 10). However, this is different from the embodiment defined by Claim 50, which defines “a workflow language that is specified in the form of annotations to source code and classes” that exchanges “information including files, documents, and/or tasks between system resources according to a set of procedural rules to generate activities at the computer as defined by the workflow definition”. Applicant respectfully submits that the use of a JavaDoc annotation tag to start a conversation is merely a way to identify a two-way communication, e.g., between one client application and one Web Service so that messages are always returned to the correct client, and that the use of annotations in Peltz simply cause a response to a request to be returned to the correct requestor. In contrast, Claim 50 defines using annotations in source code to generate activities at the computer as defined by the workflow definition in order to perform a set of steps or actions that occur in a particular order, and in response to specific conditions, in order to accomplish a task.

Additionally, as defined by Claim 50, “workflow language extends the source code with a plurality of workflow constructs, including constructs for defining parallel processing of a workflow and separate workflow branches therein”. These activities include workflow constructs “for defining parallel processing of a workflow and separate workflow branches therein”. As explained in the specification, in an embodiment, these structured activities allow the system to call an operation on a control, call a piece of Java code, or a control to call back the workflow. Peltz instead provides an example of using annotations to identify a two-way communication. Annotations, as described in Peltz, do not appear to include workflow constructs defining parallel processing of a workflow and separate workflow branches. This is because the use of annotations described in Peltz have no direct effect on the operation of the

source code they annotate. As such, the JavaDoc annotation tags disclosed in Peltz do not render obvious the limitations defined by Claim 50.

In view of these comments, Applicant respectfully submits that Claim 50 is neither anticipated by, nor obvious in view of the cited references when considered alone or in combination, and reconsideration thereof is respectfully requested.

#### **Claim 57 and 75**

The comments provided above with respect to Claim 50 are hereby incorporated by reference. Claims 57 and 75, while independently patentable, recite limitations that, similarly to those described above with respect to Claim 50, are not rendered obvious by the cited references. For similar reasons as provided above with respect to Claim 50, Applicant respectfully submits that Claims 57 and 75 are likewise neither anticipated by, nor obvious in view of the cited references, when considered alone or in combination, and reconsideration thereof is respectfully requested.

#### **Claims 51-56 and 58-63**

Claims 51-56 and 58-63 depend from and include all of the features of Claims 50 and 57. Claims 51-56 and 58-63 are not addressed separately but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and the comments provided above. Reconsideration thereof is respectfully requested.

#### **IV. Request for Interview**

Applicant respectfully submits that the claims, as amended, are now allowable. However, if the next Office Action is not an allowance of the claims, Applicant requests a telephonic interview with the Examiner prior to issuance of the next Office Action in order to expedite the prosecution of this application.

#### **V. Conclusion**

In view of the above amendments and remarks, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

Applicants believe that no fee is due with this communication. However, the Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this reply, including any fee for extension of time which may be required.

Respectfully submitted,

Date: March 12, 2009

By: /Adam T. Hipp/  
Adam T. Hipp  
Reg. No. 60,334

Customer No. 80548  
FLIESLER MEYER LLP  
650 California Street, 14th Floor  
San Francisco, California 94108  
Telephone: (415) 362-3800

- 12 -

Application No. 10/784,375  
Attorney Docket No. ORACL-01389US1  
M:\ahipp\wp\ORACL\1389US1\1389US1\_ROA\_03052009\_2.doc